

[Table of Contents:](#)

GTBay Data Types

[Data Types](#)

GTBay Constraints

[Constraints](#)

Task Decomposition & Abstract Code

[Login Form](#)

[Registration Form](#)

[Main Menu / Navigation Bar](#)

[New Item Form](#)

[Search Item and Results Form](#)

[Item Description](#)

[Update Description](#)

[Item Ratings](#)

[Auction Results](#)

[Category Report](#)

[User Report](#)

GTBay Test Cases

[Test Cases](#)

Data Types:**User**

Attribute	Data Type	Nullable
username	String	Not Null
password	String	Not Null
first_name	String	Not Null
last_name	String	Not Null
position	String	Not Null

Item

Attribute	Data Type	Nullable
item_name	String	Not Null
description	String	Not Null
category	String	Not Null
condition	Integer	Not Null
returnable	Boolean	Not Null
starting_bid	Decimal(12,2)	Not Null
min_sale_price	Decimal(12,2)	Not Null
auction_len	Integer	Not Null
getitnow_price	Decimal(12,2)	NULL

Rating

Attribute	Data Type	Nullable
item_name	String	Not Null
num_of_stars	Integer	Not Null

comment	String	Not Null
date_and_time	Datetime	Not Null

Bidinfo

Attribute	Data Type	Nullable
bid_amont	Integer	Not Null
time_of_bid	Datetime	Not Null

Constraints:

GTBay User

- Users who are new to GTbay must register first.
- Users who have an existing GTbay account will not be able to register.

GTBay Item

- The ItemName should be unique for each item that belongs to the same user.
- Only the listing user could edit the description of items

GTBay Bid

- Bid amount must be higher than starting bid.
- Bid amount must be at least \$1 above the current highest bid.
- Bid amount should not be higher than get it now price if the get it now price has been set.
- Sale price must be higher than Minimum Sale Price.
- All prices fields only allow two decimal places.
- A user can rate an item only once.

Task Decomposition & Abstract Code

Login Form

Task: Login



Task Decomposition

Lock Types: Read-only verification on [User](#) table

Number of Locks: Single

Enabling Conditions: None

Frequency: High

Consistency (ACID): Not critical, order is not critical.

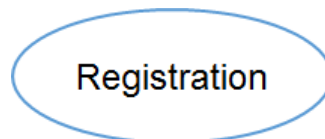
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract code

- User enters *Username* and *Password* input fields and click **Login** button
- If data validation is successful for both *Username* and *Password* input fields, then:
 - When **Login** button is clicked:
 - If User record is found but `User.password != '$Password'`:
 - Go back to Login form, with error message.
 - Else:
 - Store login information as session variable '\$UserID'.
 - Go to View Profile form.
- Else *Username* and *Password* input fields are invalid, display Login form, with error message.

Registration Form

Task: Registration



Task Decomposition

Lock Types: Add new users to the system

Number of Locks: Single

Enabling Conditions: Triggered by clicking **Register** button in the Login Form

Frequency: Low

Consistency (ACID): Not critical, order is not critical

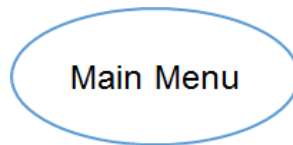
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract code

- User clicks on the **Register** button in the Login Form
- User will then be redirected to the Registration Form
- User enters *FirstName*, *LastName*, *Username*, *Password* and *Password Confirmation* then clicks on **Register** button
- User will be redirected to the Main Menu after account being created

Main Menu / Navigation Bar

Task: Main Menu



Task Decomposition

Lock Types: Lookup **User** Name, all are Read-only.

Number of Locks: Single

Enabling Conditions: Triggered by successful login.

Frequency: User Detail and Menu Options have the same frequency.

Consistency (ACID): not critical, order is not critical.

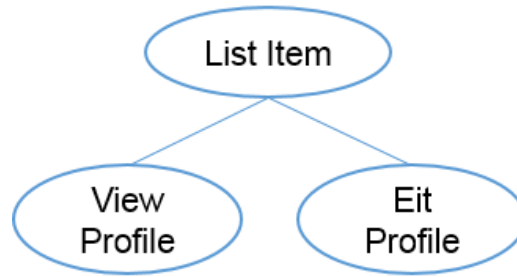
Subtasks: Mother Task is not needed. No decomposition needed.

Abstract code

- Show **List Item**, **Search for Item**, and **Logout** buttons for both Regular User and Administrative User. Besides, show **View Category Report** and **View User Report** buttons for Administrative User.
- Upon:
 - Click **List Item** button - Jump to the **List Item** task.
 - Click **Search for Item** button - Jump to the **Search for Item** task.
 - Click **View Category Report** button - Jump to the **View Category Report** task.
 - Click **View User Report** button - Jump to the **View User Report** task.
 - Click **Log Out** button - Invalidate login session and go back to the Login form.

New Item Form

Task: List Item



Task Decomposition

Lock Types:

- Lookups of Category and Condition lists
- Edits of new item information

Number of Locks: Several different schema constructs are needed.

Enabling Conditions:

- All two are enabled by a user's login
- Edits of new item are enabled by edit request

Frequency: High

Consistency (ACID): Not critical, order is not critical

Subtasks: All tasks must be done. Mother task is required to coordinate subtasks. Lookup done first followed by any number of edits.

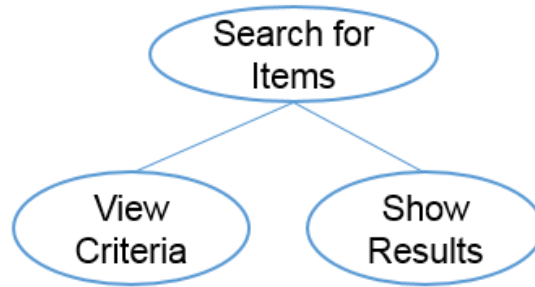
- 'View Profile' is read-only
- 'Edit Profile' updates [Item](#) only

Abstract code

- User clicks on the **List My Item** button in the Main Menu/ Navigation Bar
- User will then be redirected to the New Item Form
- Run the **View Profile** subtask: **View Profile. Populate Category and Condition dropdowns.**
- Run the **Edit Profile** subtask: User enters *Item Name, Description, Category, Condition, Start auction bidding price, Minimum sale price, Auction end time, returnable and Get it now price* (optional)
- If click **List My Item** button: **Update View Item Detail**
- If click **Cancel** button: user will be redirected to the Main Menu and selects next action from choices in Main Menu

Search Item and Results Form

Task: Search for Item



Task Decomposition

Lock Types: Read-only lookups of Items and BidInfo, Category list and Condition list

Number of locks: Several different schema constructs are needed.

Enabling Conditions: The task is enabled by a user's login or a user's lookup

Frequency: High. All two have the same frequency

Consistency: Not critical.

Subtasks:

All tasks must be done, but can not be done in parallel. Mother task is needed to coordinate the subtasks. **View Criteria** and **Show Result** are read-only. Order is necessary. **View Criteria** should be followed by **Show Result**.

Abstract code

- User clicks **Search for Items** button from **Main Menu**
- Run the **View Criteria** subtask. **Populate Category and Condition dropdowns.**
- If **Cancel** is clicked, go back to **Main Menu**
- User input searching criteria. User input *Keyword, Minimum Price, Maximum Price, Condition at least.*
- If user clicks **Search** button - Jump to the **Show Result** subtask:
 - Find the current **Item** for sale using the **Item**.ItemName; Display Item ID and Item Name;
 - Find the current **BidInfo** using the **Item**.ItemName; Display Current Bid, High Bidder, Auction Ends that match the criteria
 - If **Item**.GetItNowPrice has been set: display **Item**.GetItNowPrice; else, do nothing.
 - Sort the results of items in order of the auction end date with the auction ending the soonest listed first
- If **Back to Search** is clicked, go back to **View Criteria** task.

Item Description

Task: Item Description



Task Decomposition

Lock Types:

- 1 read-only lookups of information for an Item
- 1 read-only lookups of information for a BidInfo
- Edits of Bid information: insertion of new bid information

Number of locks: Several different schema constructs are needed.

Enabling Conditions:

- All four are enabled by a user's login
- Edits of Bid information are enabled by separate edit request

Frequency:

- High- Two read-only lookups have the same high frequency
- Moderate- insertion of new bid information
- Low- updates of the bid information if the user purchase the item immediately using Get It Now Price

Consistency: It is critical for edits part:

- clicking on "Get It Now!" button NEEDS to be done at once to purchase the item successfully, otherwise the tables could become inconsistent
- bidding for the item NEEDS to be done at once, otherwise the tables could become inconsistent

Subtasks: **View Item Details** and **View Bid Details** must be done. **Get it Now** and **Add Bid** are optional. Tasks of viewing and tasks of editing cannot be done in parallel:

- Lookup done first followed by any number of edits;
- Update of the highest price should not be done before insertion of bid amount;

Mother task is required to coordinate subtasks:

- '**View Item Details**' is read-only
- '**View Bid Details**' is read-only
- '**Get It Now**' updates BidInfo only
- '**Add Bid**' updates BidInfo only

Abstract Code

- User clicked on the specific item name from **Search Results**:
- Run the **View Item Details** subtask: query for information about the item and their profile where \$Item ID is the ID of the current item using the system from the HTTP Session/Cookie:

- Find the current **Item** using the **Item.ItemName**; Display Item ID and Item Name;
- Find the current **Item** using the **Item.ItemName**; Display Item Description, Category, Condition, Returnable and AuctionEndTime.
 - If the Get It Now price has been set:
 - Display GetNowPrice.
 - Else:
 - Do nothing.
- Run the **View Bid Details** subtask: query for information about the user, bid and their profile:
 - Find the current **BidInfo** using the **Item.ItemName**; Display BidInfo BidAmount and TimeOfBid;
 - Find the current **User** using the **BidInfo**; Display **User** UserName


While no button is pushed, do nothing.

While a button is pushed, do the following:

- Click **View Ratings** button- Jump to the **Item Rating** task.
- Click **Edit Description** button- Jump to the **Edit Description** task.
- User input *Your bid*, and click **Bid On This Item** button- Jump to the **Add Bid** subtask:
Update Bid.View Bid Details;
 - If the Get It Now price has been set:
 - Click **Get It Now!** button- Run the **Get It Now** subask: **Update Bid. View Bid Details.**
 - Else:
 - do nothing
- Click **Cancel** button- Jump to the **Search Results** for current user.

Update Description

Task: Edit Description



Edit Description

Task Decomposition

Lock Types: Edits of item description

Number of Locks: Single

Enabling Conditions: Edits of item description is enabled by edit request. Edits are enabled by a user's login who listed the certain item.

Frequency: Relatively low

Consistency (ACID): not critical, order is not critical

Subtasks: Mother Task is not needed. No decomposition needed.

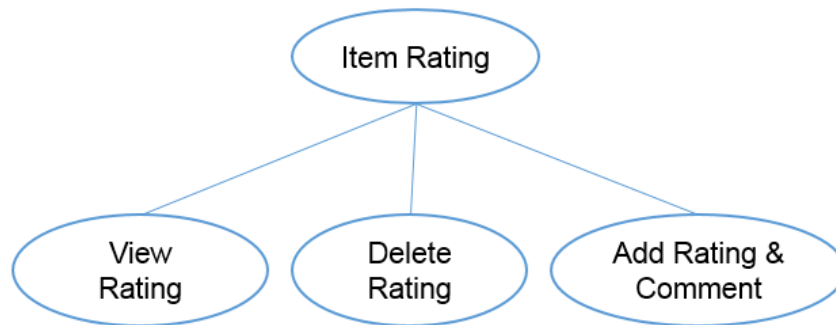
Abstract Code

If the current user is also the listing user:

- User clicked on **Edit Description** from Item Description form; run the **Update Description** task:
 - User inputs *description*, then clicks **Save** button- Update **Description. View Item Details**
 - Click **Cancel** button- Jump to the **View Item Details** for current user.

Item Ratings

Task: Item Rating



Task Decomposition

Lock Types:

- Read-only lookup of ratings made by users for this item
- Insertion of the rating of item by the current user
- Deletion of the rating posted by the current user

Number of Locks: Several different schema constructs are needed.

Enabling Conditions: All three are enabled by a user's login

Frequency:

- High frequency for **View Rating** Task
- Moderate frequency for **Add Rating and Comment** Task
- Low frequency for **Delete Rating** Task

Consistency: Consistency is critical

- When the current user deletes the rating, the page information should be updated in time.
- Adding rating and comment NEEDS to be done at once, otherwise the tables could become inconsistent

Subtasks: View Rating must be done. Delete Rating and Add Rating and Comment are optional. All the tasks cannot be done in parallel:

- Lookup done first followed by any number of edits;
- If a user has rated the item before, adding rating and comments should not be done before deleting the former rating;

Mother task is required to coordinate subtasks:

- **'View Rating'** is read-only
- **'Delete Rating'** and **'Add Rating and Comment'** update Rating.

Abstract code

- Click **View Ratings** button on Item Description form, run the **View Rating** subtask:
query for rating information of item:
 - Find the current **Item** using the **Item.ItemName**; Display Item ID and Item Name;
 - Find the current **Rating** using **Item.ItemName**. Sort and display the ratings made by users for this Item (sort by datetime that rating was made)
 - Calculate and display average Num of Stars made by users for this item
 - Find the **User** that made the most recent rating using **Rating**. Display Username of **User**, NumberOfStar, Comment and DateAndTime of **Rating**.
 - If the current user has rated this item before:
 - Find the current **User** using **Rating**. Display Username of current **User**, NumberOfStar, Comment and DateAndTime of **Rating**.
 - Else:
 - Do nothing.
- If the current user has rated this item before, when a User clicks **Delete My Rating** button, jump to the **Delete Rating** subtask:
 - **Delete** the NumOfStars, Comment, DateAndTime of Rating made by the current user from database. **Update Rating.View Rating**.
- User rates items from 0 to 5 stars and enter comments into the field. Click **Rating This Item** button, then run the **Add Rating and Comment** subtask:
 - If it is the first time for the current user to rate the item:
 - **Add** the NumOfStars, Comment, DateAndTime of Rating made by the current user from database. **Update Rating.View Rating**.
 - Else:
 - Pop out an error window with error message ""You can rating the item only once".
- Click **Cancel** Butt
- on, return user to Item Description form.

Auction Results



Task: View Auction Results

Lock Types: Read-only lookups of Item ID, Item Name, sale price, Username of Winner, the Date and Time that the auction ended.

Number of Locks: Several different schema constructs are needed.

Enabling Conditions: enabled by clicking view Auction Result button on Main menu.

Frequency: High

Consistency: not critical

Subtasks: mother task is needed

Abstract code

- User clicks **View Auction Results** button on Main Menu. Run the **View Auction Result** task:
 - Find **Item**, **BidInfo** and **User** according to items whose auction have ended and users who have bid for the items.
 - Display the winner information: display Item ID, Item Name, Sale Price, Username of Winner and AuctionEndTime on Screen.
 - For each item in query result:
 - If bid info for the item is not null:
 - Sort by the bid amount made by different users to get the highest price and its username.
 - If the highest price < minimum sale price:
 - Set the Sale price and Winner of item to blank
 - Else:
 - Set the highest price and Username as the final sale price and Winner of this item.
 - If bid info for the item is null:
 - Set the sale price and winner to blank.
 - Display the sale price and winner of item on screen
- Click **Done** Button, return user to Main Menu.

Category Report

Task: View Category Report



Task Decomposition

Lock Types: Read-only lookups of Items and BidInfo

Number of locks: Several different schema constructs are needed

Enabling Conditions: administrative user can trigger this task by clicking the **View Category Report** button on Main Menu

Frequency: High

Consistency: Not critical

Subtasks: Mother task is not needed

Abstract code

- When Administrative User clicks **View Category Report** button on Main menu, run the **View Category Report** task:
 - For each category:
 - Calculate the total number of items, minimum price, maximum price and average price

- Display Category and the calculation results above.
- If **Done** is clicked, go back to **Main menu**

User Report

Task: View User Report



Task Decomposition

Lock Types: Read-only lookups of Items and BidInfo

Number of locks: Several different schema constructs are needed

Enabling Conditions: Administrative User could trigger **View User Report** by clicking the **View User Report** button on **Main Menu**

Frequency: High, consistent across the task

Consistency: Not critical

Subtasks: no mother task is needed

Abstract code

- When Administrative User clicks **View User Report** button on **Main menu**, run the **View User Report** task
 - For each user:
 - Calculate the total number of items listed, sold, purchased and rated.
 - Display Username and the calculation results above.
- If **Done** is clicked, go back to **Main menu**

Test Cases:

TC#	Area	Test Case Description
L000	Login	
L001		Login page entry (username, password) credentials
L002		Valid credentials (username, password) accepted
L003		reject invalid (username, password) when user does not exist
L004		reject invalid (username, password) when user exists, password invalid
R000	Register	
R001		Registration page entry (first name, last name, username, password, confirm password) credentials
R002		Valid registration when all required registration information are entered (first name, last name, username, password, confirm password) and Password and Confirm Password are the same
R003		Reject invalid registration when one or more required information is missing (either username, password, full name or email)
R004		Reject invalid registration when Password and Confirm Password are different
NI000	New Item	
NI001		User will get meaningful feedback if some of the input is not correct (eg. required field is not provided, invalid input format).

NI002		Category list is correctly populated from database.
NI003		User input data is correctly stored in database.
S000	Search Item	
S001		Search result is still for auction.
S002		Search result matches the search criteria.
VC000	View Category Report	
VC001		View category report tab will not shown to regular user.
VC002		Category report is shown correctly.
VU000	View User Report	
VU001		View user report tab will not shown to regular user.
VU002		User report is shown correctly.
VI000	View Item	
VI001		Item info(item name, description, category, condition..)stored correctly
VI002		Item info(item name, description,category, condition...)displayed correctly
VI003		Latest 4 bids displayed correctly
B000	Bid	
B001		Bid info entry (bid amount)
B002		Valid bid amount accepted
B003		Reject invalid bid amount
G000	Get it Now	
G001		User successfully bid the item
G002		Auction for the item will end, reject other user's bid
ED000	Edit Description	
ED001		This option only visible for the current user who is also the listing user

ED002		Description of item entry
ED003		Description of item stored correctly
VAR000	View Auction Results	
VAR001		User, bid and item info(item id, username, bid amount,date and time) is stored correctly
VAR002		The sale price and winner is calculated correctly
VAR003		Auction result is displayed correctly
VR000	View Rating	
VR001		Rating info of item (num of stars, date and time, and comment) is stored correctly
VR002		All rating info related to the current item is sorted and displayed correctly
DMR000	Delete My Rating	
DMR001		This command is only visible for the current user who have made the rate for item
DMR002		The rating info made by user will be successfully delete from database and the page is refreshed.
ARC000	Add Rating & Comment	
ARC001		Rating info(num of stars and comment) input
ARC002		Rating info made by user who never rate this item before accepted
ARC003		Reject rating info input of user who made multiple times of rating and never delete previous rating)